CARNEGIE-MELLON UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SPICE PROJECT

Oil: The Spice Ascii Editor

Keith Wright

16 Aug 84

This manual includes sections from the PERQ Systems Manual, *The Editor*.

Table of Contents

1 Introduction

Oil is an editor developed by the Spice project for use on the Perqs. OIL incorporates features of other text editors, and it resembles the Spice interim editor, PEPPER. By accessing the screen through SAPPHIRE, Oil takes advantage of the large virtual storage capacity of Accent, and it allows you to position the cursor with a mouse as well as with the keyboard. This document describes the basic features and commands of Oil.

2 Overall View

2.1 The Display

Oil provides two kinds of windows: a prompt window and a text window. The prompt window shows information on the current state of the editor, with error messages, questions, and the contents of the kill buffer[1]. One or more text windows may be displayed beneath the prompt window. A text window is made up of three areas:

- The text you are working with or the blank space where the text should be.
- The status bar, which is the horizontal section set off above the text. It contains the name of the file and the *thumb bar*, which causes the cursor arrow to move in a more dramatic way through the text than with the keyboard alone.
- The scroll bar, which is the vertical section to the left of the text. The *scroll bar*, as its name suggests, enables you to scroll through the text. For more information on thumbing and scrolling, see Section 3.5.

2.2 The Pencil and the Cursor

A short vertical bar called the *pencil* marks the place in the text where the editing will take place. It is always between two characters, which means that most commands act on the characters immediately preceeding or following the pencil. You cannot move the pencil outside of the visible text.

A little black arrow known as the *cursor* is controlled by the mouse next to your keyboard. The cursor can be moved to any position on the screen. The cursor can be used to move the pencil to any position in the text buffer. However, the pencil, not the cursor, affects your editing.

2.3 The Mouse

The bit pad or tablet lying next to your keyboard uses the mouse to simulate the movement of the

---

[1] See section 2.4 for information about the kill buffer.

cursor on the screen. The mouse can be picked up and relocated on the bit pad without disturbing anything on the screen, and in this way you can get the cursor into a new position without running out of space on the pad. Note that the mouse is one convenient way of invoking editor commands. It substitutes for the Control (**Ctrl**) key on the keyboard.

To put the pencil into the same position of the cursor-arrow in the text, move the cursor to the spot you want and press the yellow button on Perq1 or the center button on Perq2. The yellow and blue buttons on the Perq1 and the right and left buttons on the Perq2 are used for thumbing and scrolling through text as well.[2]

2.4 The Kill Ring

Every time text is deleted it is saved in a kill buffer. Oil uses a kill ring consisting of 8 buffers. Each kill buffer contains text deleted by one or more consecutive kill commands. When text is deleted the oldest buffer in the kill ring is overwritten. Only the most recent kill buffer is displayed in the prompt window. The **yank-buffer** command inserts the displayed kill buffer into the file at the pencil. Earlier buffers on the ring can be accessed with the **pop-kill-ring** command. [3]

3 Commands

Oil is a modeless editor, which means that pressing a key will always have the same effect. Pressing ordinary alphanumeric keys causes the appropriate character to be inserted into the text. **Control x** (**Ctrl-x**) serves as a meta or escape key; Crtl-x and an accompanying key should be counted as a single compound stroke.

To invoke a command, hold down either the Ctrl key or the mouse key (yellow on Perq1, center on Perq2) while striking the appropriate alphanumeric key listed in parentheses after the command name.

This section describes Oil's commands, divided according to function: invoking the editor, leaving the editor, inserting text, deleting text, changing position in the text, searching for and replacing text, selecting text and transcribing text. There are some miscellaneous commands that may be useful at the end of the section, and a command summary that follows it.

3.1 Invoking Oil

Oil can be invoked by typing "edit" or "edit /pathname/<filename>" to the shell. The given

---

[2] See section 3.5, "Changing Position", for details on thumbing and scrolling.

[3] See section 3.8, "Miscellaneous Commands", for more information on the pop-kill-ring command.

pathname is used to locate a file with the current searchlist in the SESAME file system. If you want to edit the filename, Oil will allow you to do so, using any of the line-editing commands. The extensions ".pasmac", ".body", ".pas", ".cmd", ".mss", or ".micro" are appended to the given name in an attempt to find the file. If there is no file by that name in your directory an empty file is created.

Oil also has an escape completion feature which allows the user to type as much of the filename as needed to distinguish the file from other files and then INS or ACC(ESC) on a Perq2. If the enough of the filename has been supplied to identify it, oil will supply the rest of the name. If the user has not typed enough of the filename Oil will list existing files beginning with the letters typed so far and wait for more information.

3.2 Leaving Oil

Typing Ctrl-x Ctrl-f (the **write-file-exit** command) will get you out of Oil. At this point the filename may be edited or one may be written if the file was not previously named. The "RETURN" key will save the filename. Hitting the "Oops" key and then "RETURN" will result in exiting Oil without writing to the file.

As you edit Oil keeps a continuous transcript of the changes you make. If any mishap occurs this transcript can be replayed, so that your work is not lost even if you do not save the changes before leaving the editor. See section 3.9 for more information.

3.3 Inserting Text

Inserted text is put in before the pencil. Unless otherwise specified, inserting a character causes the pencil to move forward.

insert-tab   **(Ctrl-TAB)**
Insert enough spaces to make the current position congruent to one modulo eight.

new-line   **(RETURN)**
Insert a carriage return to split the line at the current position.

new-line-and-indent   **(LF)**
Split the line and indent the second half the same amount as the first half.

open-space   **(Ctrl-o)**
Split the current line, but do *not* advance the pencil. That is, stop the pencil at the end of the first half.

quote-character	**(Ctrl-q)**
	Treat the following character as a character to be inserted and not as a command. This command allows you to insert control characters in the file.
yank-buffer	**(Ctrl-y)**
	Insert the contents of the kill buffer at the current position. This command provides a way to transfer or copy text. Note that text deleted from one window may be yanked into another.
set-parameters	**(Ctrl-P)**
	Prompt for one of the following parameters:

left margin	type l. User is prompted for a value greater than 0. The default is 0.
right margin	type r. User is prompted for value. The default is 70.
tab	type t. User is prompted for a value. The default is 8 (tabs are set every 8 spaces across the screen).
word wrap	type w. This reverses the ON/OFF setting. The default is OFF. When the pencil reaches the right margin it will go to the next line taking with it the last whole word.
status	type s. This shows the values that are in effect for the above parameters.

3.4 Deleting Text

All of the following commands can be reversed by the **yank-buffer** command.

delete-next-character	**(Ctrl-d)**
	Delete the character to the right of the pencil.
delele-previous-char	**(BACK SPACE)**
	Delete the character to the left of the pencil and move the pencil back.
kill-to-end-of-line	**(Ctrl-k)**
	If the pencil is not at the end of a line then delete all characters in the line after the pencil. If the pencil is at the end of the line then append the following line by deleting the carriage return.

kill-to-start-of-line **(OOPS)**
If the pencil is not at the beginning of the line then delete all characters from the beginning of the line to the pencil. If the If the pencil is at the start of the line then append the line to the preceding line.

3.5 Changing Position

You can change your focus of attention in the text by using either the mouse or the keyboard, or both.

3.5.1 Mouse Positioning: Thumbing

The file being edited is linearly mapped onto the thumb bar above the text, with the part of the file on the screen before you represented by a black square. The left end of the bar represents the top of the file and the right end the bottom. If you point the cursor arrow at the appropriate part of the bar and press the yellow button on Perq1 or the center button on Pcrq2, Oil will "thumb" through the file to the place you indicated. This is a great way to rapidly access distant parts of a file.

3.5.2 Mouse Positioning: Scrolling

To scroll your text sequentially up or down, move the cursor arrow into the scroll bar to the left of the text and even it up with a line of the text. Press the yellow button on Perq1 or the right button on Perq2 to move that line to the top of the text window. Press the blue button on Perq1 or the left button on Perq2 to cause the line at the top of the screen to move to the position of the cursor.

3.5.3 Keyboard Commands

forward-character **(Ctrl-f)**
Move the pencil forward one character. If it is at the end of a line move to the beginning of the next line.

backward-character **(Ctrl-b)**
Move the pencil back one character. If it is at the start of a line move to the end of the previous line.

begining-of-line **(Ctrl-a)**
Move the pencil to the beginning of the current line.

end-of-line **(Ctrl-e)**
Move the pencil to the end of the current line.

previous-line   **(Ctrl-p)**
Move the pencil to the same column of the previous line, unless
that line is too short. If the line is to short the pencil moves
to the end of the line, but remembers the column it is supposed
to be in so that if it is moved to a longer line it will return
to its original column. Holding down the "p" key can be a
convenient way to scroll down.

next-line   **(Ctrl-n)**
Move the current position to the same column of the next line,
unless that line is too short. If the next line is too short
move to the end of the line. The "true" column is remembered
just as for the previous-line command. Holding down the "n" key
can be a convenient way to scroll up.

up-page   **(Ctrl-V)**
Scroll the text down the height of the window. The pencil remains
in the same row and column on the screen.

down-page   **(Ctrl-v)**
Scroll the text up the height of the screen. The pencil remains in
the same row and column on the screen.

top-of-window   **(Ctrl-,)**
Move the pencil to before the first character of the first line in
the window.

bottom-of-window   **(Ctrl-.)**
Move the pencil to before the first character of the last line in
the window.

top-of-file   **(Ctrl-<)**
Move the pencil to before the first character of the first line in
the book.

bottom-of-file   **(Ctrl->)**
Move the pencil to just before the end of the book.

forward-word   **(Ctrl-F)**
Move the pencil to the end of the next word.

backward-word   **(Ctrl-B)**
Move the pencil to the beginning of the previous word.

3.6 Searching for and Replacing Text

This group of commands allows you to move through a file by doing a search for a string. The search and replace commands cause Oil to prompt for strings it is to search for. The searches are not case sensitive. The previous string searched for is displayed after the prompt in the prompt window. This previous string may be edited using all of the ordinary editing commands. The previous string will be deleted when a character is typed. This makes it very easy either to make small changes to the old string or to enter an entirely new one.

NOTE: A carriage return is not a character that can be entered into a search string–it will end the string.

search-forward   **(Ctrl-s)**
Prompt for a string and search for a match in the current text. The search starts at the current position of the pencil and proceeds to the end of the text. The pencil is left at the end of the first occurance of the given string in the text if one is found; otherwise it is not moved.

reverse-search   **(Ctrl-r)**
Prompt for a string and search for a match in the current text. The search starts at the current position of the pencil and proceeds backwards to the beginning of the text. The pencil is left at the beginning of the first occurance (counting backwards) of the given string in the text if one is found; otherwise it is not moved.

replace   **(Ctrl-R)**
Prompt for a search string and a replacing string. It repeatedly searches for the given string, displays each occurance as it is found, and asks you whether or not the string should be replaced. At this point you may type:

- INS to replace the string and continue the search for further occurances.
- a blank to leave that occurance as it is but to continue the search for others,
- TAB to replace the occurance but cancel the search, or
- DEL to abort the command.

3.7 Manipulating Windows

These commands permit you to examine and edit up to nine files at one time. However, even

when more than one text window is present there is only one prompt window and one set of kill buffers. The same commands that move text within one file also transfer text between files.

visit-file   **(Ctrl-x Ctrl-v)**
Make a new window by splitting the current one in half. If you are in a small window it may not be possible to create a new one even though there are larger widows. In this case just move to a larger window and repeat the visit-file command. The command will prompt for the name of a file to be edited in the new window, and it will give a suggested title that may be edited as usual. The suggested file for each successive visit-file command will be the list of file names given in the shell command with which the editor was invoked, taken in order.

If instead of a file name an equal sign (=) is given, the current window is split. The two windows view the same file; changes made in one window will appear in both, but the two windows can be scrolled independently.

If an equal sign followed by a file name is given, the transcript of that file is replayed. See section 3.11, "Transcribing Text", for more information on file transcripts.

delete-window   **(Ctrl-x Ctrl-d)**
Remove the current window. The editor prompts for the filename to write the updated text into if one was not previously given. Any existing filename may be edited at this point. Thus to discard the changes which have been made simply type "Ctrl-x Ctrl-d OOPS RETURN". To help ensure that the wrong window is not deleted by accident, the window which is about to be deleted is turned to reverse video (white on black) when the prompt is displayed. Type DEL at that time to abort the delete command. If the last window is deleted, a new one is immediately created.

next-window   **(Ctrl-x Ctrl-n)**
Move the pencil to the next window, in the order of creation. If the pencil is in the very last window, move it to the first.

3.8 Selecting Text

Portions of the text may be *selected* and displayed underlined. Commands are provided to delete or copy selected text. This is often easier and safer than using the line and character deletion commands, particularly for large blocks of text.

start-selection	**(Ctrl-[)** The current position of the pencil is marked as the start of the selection. If the end of selection has already been marked further down in the text, then everything between the beginning and the end will be underlined. If the end of the selection has not been marked there will be no visible effect, but the start is marked.
end-selection	**(Ctrl-])** The current position of the pencil is marked as the end of the selection. If the end of selection has already been marked further down in the text, then everything between the start and the end will be underlined. If the start of the selection has not been marked there will be no visible effect, but the end is marked.
select	**(left button)** If the left mouse button is pressed once the character pointed to by the cursor arrow (not the pencil) is selected. If it is pressed again, the word is selected; if yet a third time the entire line is selected.
modify selection	**(right button)** If the right cursor button is pressed the end of the selection which is closed to the cursor arrow is moved to the place pointed to by the arrow.
delete selection	**(Ctrl-")** The currently selected text is deleted. A copy of it is placed in the kill buffer.
insert selection	**(Ctrl-')** The currently selected text is inserted at the pencil.
fill selection	**(Ctrl-j)** To fill the lines of the current selection within the present margin setting. You can issue this command from anywhere in the file; the selection will fill regardless of where the cursor is.

3.9 Transcribing Text

If some catastrophy causes the editor, the operating system, the computer or the power supply to suddenly quit when you have made many changes to a file but have not yet saved them, all is not lost. The editor continuously maintains a transcript of all of the editing commands given to each

file. This transcript can be replayed and will usualy contain almost all of the work you have done. (The last few changes may not have been written to the transcript.)

For example, if you are editing a file called "Foo.pas", the transcript is maintained in the file Foo.pas+. NOTE: Do not start to edit the original file before replaying the transcript.

To replay this transcript start the editor and type "foo.pas -replay" in response to its prompting for a title. The editor will then read in the file Foo.pas. It will then display the first change which was made to the file in the prompt window. If you type INS the change will be made and the next change will be displayed. You may step through the changes one at a time in this way. If for some reason you decide you do not want to continue playing the transcript hit DEL. If you become confident or bored type an exclamation point to replay the remainder of the transcript.

After the transcript is done the book is as it was originally, it would be wise to save the changes at this point.

If for some reason you make changes to a file and save them, and then decide you didn't want them made, all is not lost. Before writing any file the editor creates a backup with the same name but with a dollar sign appended. Thus, if you make changes to foo.pas you will find the original Foo.pas under the name Foo.pas$. (To use Oil without having a backup made invoke Oil with the -Nolog switch.)

3.10 Miscellaneous Commands

upper-case word   **(Ctrl-U)**
Make the next word upper-cased.

lower-case word   **(Ctrl-L)**
Make the next word lower-cased.

transpose-characters   **(Ctrl-t)**
Swap the two characters preceding the pencil.

pop-kill-ring   **(Ctrl-Y)**
Pop the kill ring. This command moves the top (displayed) kill buffer into the last postion of the kill ring. The second kill buffer then becomes the first, and is displayed in the prompt window. There are eight buffers in the kill ring, and so eight pop-kill-ring commands will return the editor to its original state.

repeat-command   **(Ctrl-u)**
Prompts user for an argument (a number followed by a CR) and then a command. Argument sets the number of times the command will be

repeated.

set-mark   **(Ctrl-@)**
A mark is set at the current position of the pencil and associated with the current window, or with a given name. Set mark takes an argument given by the enter-argument command. This argument is alphanumeric and is interpreted as the name of the mark. If no name is given, the mark is the annonymous mark for the current window.

go-to-mark   **(Ctrl-xx)**
The pencil is swapped with a mark. Go to mark takes an alphanumeric argument via the enter-argument command. This is interpreted as the name of the mark to go to. It need not be in the current window. If no name is given the annonymous mark in the current window is used.

save keystrokes   **(Ctrl-left parenthesis)**
Save the following set of keystrokes.

end-save keystrokes   **(Ctrl-right parenthesis)**
Save set of keystrokes up to this mark, that is, stop the save-keystrokes command now.

replay keystrokes   **(Ctrl-*)**
Replay the keystroke string just created by save-keystrokes and end-save-keystrokes.

scroll-down-cursor   **(Ctrl-x Ctrl-!)**
This can be used instead of your mouse. Scroll the text line containing the cursor to the bottom of the window.

scroll-up-cursor   **(Ctrl-!)**
This can be used in place of your mouse. Scroll the text line containing the cursor to the top of the window.

cursor-left-column   **(Ctrl-X Ctrl-!)**
Move the cursor to the far left column.

4 Command Summary

```
The commands use Emacs bindings.
```

Ctrl-f	forward character	Ctrl-TAB	insert spaces
Ctrl-b	backward character	RETURN	insert <CR>
Ctrl-F	forward word	DEL, Ctrl-g	abort previous command
Ctrl-B	backward word	LF	new line and indent
Ctrl-a	begin line	Ctrl-o	open space
Ctrl-e	end line	Ctrl-q	quote character
Ctrl-p	previous line	Ctrl-y	yank kill buffer
Ctrl-n	next line	Ctrl-d	delete character
Ctrl-V	up page	Ctrl-D	delete word
Ctrl-v	down page .	OOPS	kill to start of line
Ctrl-,	top window	Ctrl-k	kill to end of line
Ctrl-.	bottom window	Ctrl-s	search forward
Ctrl-<	top of file	Ctrl-r	reverse search
Ctrl->	bottom of file	Ctrl-R	global replace
Ctrl-@	set mark	Ctrl-x Ctrl-x	swap mark with pencil
Ctrl-'	insert selection	Ctrl-"	delete selection
Ctrl-[	begin selection	Ctrl-]	end selection
Ctrl-j	fill selection		
Ctrl-(	save keystrokes	Ctrl-)	end saving
Ctrl-*	replay keystrokes	Ctrl-!	line to top of window
Ctrl-x!	line to bottom	Ctrl-X Ctrl-!	column to left of window
Ctrl-t	transpose	yellow	move cursor
Ctrl-Y	pop kill ring	(centre)	scroll up
Ctrl-x Ctrl-v	make new window	blue	-
Ctrl-x Ctrl-d	delete window		scroll down
Ctrl-x Ctrl-f	exit editor	white	select char,word,line
Ctrl-x Ctrl-s	save file	(right)	scroll down
Ctrl-x Ctrl-n	go to next window	green	extend selection
Ctrl-u	repeat command	(left)	scroll up
Ctrl-U	upper case word	Ctrl-	lower case word
Ctrl-P	set parameters:		

```
                    l   left margin
                    r   right margin
                    t   tabs
                    w   wrap mode
                    s   status
```

4.1 The Key Bindings

Oil uses SAPPHIRE to get commands from the keyboard. The key translation facilities of SAPPHIRE determine the binding of the keys to their commands, but this can be changed to suit your individual needs. The *oil.keytran* file contains all of the key definitions. You can use the Key Translation Compiler to alter the key definitions, and there is no need to recompile the editor in

order to redefine the keys. See the *Spice Commands and Utilities Manual* for more information on the Key Translation Compiler.

References

[Rashid,   Rashid, R. F.
 Robertson]   Robertson, G. B.
                    *Accent: a communicaton oriented network operating
                                    system kernel*
                 December, 1981.

[Wright 83]   Wright, Keith
                    Oil: The Spice Ascii Editor
                 July, 1983.

[Wright,
 Schriver]   Wright, Keith
                 Schriver, Karen A., *revisor*
                    Oil: The Spice Ascii Editor
                 July, 1983.